METHOD FOR CONTROLLING TRAP GENERATION OF SNMP

BACKGROUND OF THE INVENTION

1.   Field of the Invention

The present invention relates to an SNMP (Simple Network Management Protocol), and more particularly, to a method for controlling trap generation of an SNMP.

2.   Description of the Background Art

One of network management protocols for a communication network management, the SNMP defines a control structure for a management-object resource (an object) on the basis of an RFC 1157 standard.

Figure 1 shows a construction of a manager and an agent adopting an SNMP in accordance with a conventional art.

As shown in Figure 1, the SNMP is adopted in the communication network consisting of a central SNMP management system (manager)10 and at least one SNMP management-object system (agent) 12.

The manager 10 serves to control a management-object resource (object) of the agent 12 through the SNMP. That is, the manager 10 outputs an object access message (GET/SET/GETNEXT) to the agent 12 and performs searching, changing, generating or deleting on the object defined in an MIB (Managed Information Base) of the agent 12.

The GET message is a message to read a data from the MIB 14, and the MET message is a message to write a data in the MIB 14. The GETNEXT

1

message is a message to read an object next to the object read by the GET message. A GETResponse is a message to transmit the data read from the MIB 14 to the manager 10 according to the GET/GETNEXT message.

Accordingly, when the manager 10 searches information, it transmits an
5 OID (Object ID) together with the object access message (GET/GETNEXT) to the agent 12, and the agent 12 transmits the GETResponse message including a value of the corresponding data together with the OID back to the manager 10. In this respect, the OID is the ID of the object, and every data is discriminated by the OID.

10 The SNMP supports a 'Trap' operation so that the agent 12 voluntarily transmits information on the object to the manager 10, without depending on the request of the manager 10.

Describing a management object behavior, access authorization and grammar structure allowable for the object existing in the MIB 14, the SNMP
15 defines a trap in the MIB 14 by relating an object to be subjected to trap generation to a condition of the trap generation. The trap is defined as follows:

ObjectName TRAP-TYPE

ENTERPRISE {enterprise name}

20 VARIABLES {variable name}

DESCRIPTION "comment"

:: = Sub OID

Accordingly, when the state (i.e., system up/down and system
25 disturbance) of the object is changed, the agent 12 voluntarily outputs a TRAP

2

PDU (Protocol Data Unit) to inform the manager 10 of the state change of the object. In this respect, the TRAP PDU includes an OID and a corresponding data value.

The network management protocol through the SNMP will now be
5 described in detail.

The manager 10 outputs an ID of the object (OID) together with the GET message to the agent 12 in order to search the state of the object of the agent 12. At this time, the OID is an ID of each object, by which every management-objected data is discriminated.

10 Upon receipt of the GET request, the agetn 12 reads a data value from the MIB 14 and transmits the GETResponse message to the manager 10. The GETResponse message includes a pair of an OID and a read data (OID and a read data) form.

The manager 10 outputs the next OID together with the GETNEXT
15 message in order to search the next object, and in response, the manager 12 transmits the GETResponse message in the same form to the MANAGER 10. This operation is repeatedly performed so as for the manager 10 to search the state of the every object.

When the manager 10 writes a data in the MIB 14, it outputs an OID of the
20 object and a SET message, and the agent searches a target data by using the OID and changes a corresponding data value.

Meanwhile, unlike the GET/GETNEXT/SET message, the trap management behavior is used for the agent 12 voluntarily report the state of the object periodically. That is, after the agent 12 relates a specific data and a trap
25 generation condition, when it comes to a predetermined cycle, the agent 12

3

outputs a trap PDU (Potocol Data Unit) to inform the manger 10 of the change in the state of the object. Also, the TRAP PDU consists of a pair of an OID and a data, the same as that of the GETResponse message.

5   However, in the SNMP standard-based MIB technique of the conventional art, the object to be subjected to the trap generation is statically defined. Accordingly, conventionally, since the trap operation is adoptable only to the object defined in the MIB, the manager is not able to add or delete a specific object as an object as necessary during the network management operation.

In addition, once a trap condition is defined in the MIB, a trap operation is applied
10   to every defined object, resulting in that the agent generates a TRAP PDU even for a object with little state change.

Thus, the agent generates a trap more than necessary and thus the manager should process numerous TRAP PDU transmitted from the plurality of agents, so that the traffic is increased. This works as a factor to degrade the
15   management efficiency in the management network using the SNMP.

SUMMARY OF THE INVENTION

Therefore, an object of the present invention is to provide a method for
20   controlling trap generation which is capable of effectively controlling trap generation by defining a trap behavior individually for an object of an SNMP MIB.

To achieve these and other advantages and in accordance with the purpose of the present invention, as embodied and broadly described herein, there is provided a method for controlling trap generation of an SNMP operated
25   between a manager and at least one agent, wherein a TrapFlag field and a Trap

4

Peer field are defined for each management-object resource (each object) in describing an MIB of an SNMP and more than two objects are correlated to define a trap generation condition.

To achieve the above object, there is provided a method for controlling trap generation of an SNMP which is operated between a manager and at least one agent, including the steps of: defining a TrapFlag field and a TrapPeer field in an MIB of an agent; setting a TrapFlag field value according to the message outputted from the manager; setting a TrapPeer field value for each object by the agent according to the Trap generation condition defined in the MIB; and generating a trap for an object according to the values of the TrapFlag field and the TrapPeer field.

To achieve the above object, there is provided a method for controlling trap generation of an SNMP including the steps of: defining a TrapFlag field and a TrapPeer field in an MIB of an agent; and generating a trap for an object according to the values of the TrapFlag field and the TrapPeer field as defined.

In the method for controlling trap generation of an SNMP, the step of generating a trap includes the sub-steps of: searching the TrapFlag field of each object when it comes to a trap generation period; checking a state of the TrapPeer field in case that the TrapFlag is in an 'ON' state; and generating a trap for a corresponding object in case that the TrapPeer is in an 'ON' state.

The foregoing and other objects, features, aspects and advantages of the present invention will become more apparent from the following detailed description of the present invention when taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are included to provide a further understanding of the invention and are incorporated in and constitute a part of this specification, illustrate embodiments of the invention and together with the description serve to explain the principles of the invention.

In the drawings:

Figure 1 shows a construction of a manager and an agent adopting an SNMP in accordance with a conventional art;

Figure 2 shows a construction of a manager and an agent adopting an SNMP in accordance with the present invention; and

Figure 3 is a flow chart of a method for controlling trap generation of the SNMP performed in the agent of Figure 2 in accordance with the present invention.

Figure 4 is a detail flow chart of a method for generating a trap in the trap generating step S4 of Figure 3 in accordance with the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Reference will now be made in detail to the preferred embodiments of the present invention, examples of which are illustrated in the accompanying drawings.

In the present invention, when a trap object is described in an MIB of the SNMP, trap-related information is additionally defined for each object. The trap-related information is defined by adding the following fields in an MIB technique document.

1. TrapFlag field

A TrapFlag field is a field representing whether a trap is to be generated or

6

not for an object described in the MIB. The TrapFlag field is set to be turned on or off by the manager.

In case that the TrapFlag field is in an ON state, the agent generates a TRAP PDU for the corresponding object, while, in case that the TrapFlag field is in an OFF state, the agent does not generate a TRAP PDU for the corresponding object.

Accordingly, when an object of which state is little changed during a network management operation is generated, the TrapFlag field of the corresponding object is set to be 'ON", so that, even though a specific object is not deleted from the management target, the same effect can be obtained.

2. TrapPeer field

A TrapPeer field is a field defining a trap generation condition for an object, which is set by 'ON' state (logic '1') or 'OFF' state (logic '0') by the agent. That is, in case that the state of an object satisfies a trap generation condition, the agent sets a TrapPeer field as the 'ON' state.

In this respect, the trap generation condition can be defined by correlating more than two objects. For example, in case that an object 'B' is greater than 'n' and an object 'C' is greater than 'm', a TrapPeer field is defined to be set as the 'ON' state. Consequently, without adding an object, the same effect can be obtained.

Accordingly, when the TrapFlag field and the TrapPeer field are all in the 'ON' state, the agent generates a TRAP PDU for the corresponding object.

The trap generation process in the SNMP will now be described.

As shown in Figure 3, the manager 20 defines a trap as shown in below

7

by correlating objects, trap generation conditions, a TrapFlag field, and a TrapPeer field (S1).

ObjectName TRAP-TYPE

ENTERPRISE {enterprise name}

VARIABLES {variable name}

DESCRIPTION "comment"

TRAPFLAG {flag value}

TRAPPEER {flag value}

:: = Sub OID

For example, objects 'A' and 'B' are set and a trap generation condition can be defined as follows.

1) A value of the object 'A' is in the range of 1~5

2) If the value of the object 'A' is greater than '3', a TRAP PDU is basically generated.

3) If the value of the object 'B' is greater than 4, a value of the TrapPeer field of the object 'A' is set to be 'ON'.

And, the manager 20 outputs a TrapFlag setting signal to the agent 22 during network management to set a TrapFlag field value (S2). That is, the manager 20 sets a TrapFlag field of an object which shows little state change ordinarily as 'OFF' so as to count it out from an object list.

At this time, the process of the transmission of the object access message (GET/SET/GETNEXT) from the manager 20 to the agent 12 and the transmission

8

of the GETresponse from the agent to the manager 20 is the same as in the conventional art, descriptions of which are thus omitted.

The agent 22 sets a value of the TrapPeer field of each object according to a trap generation condition as defined during network operation (S3).

Thereafter, when the report period comes, first, the agent 22 generates a trap for each object according to the values of the TrapFlag field and the TrapPeer field (S4). That is, as shown in Figure 4, first, the agent 22 searches the TrapFlag field of the object 'A' to check whether a corresponding TrapFlag is in an 'ON' state (ST11).

If the TrapFlag of the corresponding object 'A' is in an 'OFF' state, even though the trap generation condition 2) is satisfied, the agent does not generate a trap. Meanwhile, in case that the TrapFlag is in an 'ON' state, it is checked whether the TrapPeer is in an 'ON' state (ST12). Upon checking, in case that the TrapPeer is in an 'ON' state, the agent 22 generates a trap for the object 'A' and performs a normal operation (ST13 and (ST14).

In this manner, a trap is generated by conditions for the more than two objects by using two fields. And, though a single agent is taken as an example for an explanation's sake in the present invention, a plurality of agents can be connected to the manager.

As so far described, according to the a method for controlling trap generation of an SNMP of the present invention, a TrapFlag field and a TrapPeer field are separately defined for each object in the MIB, and more than two objects are correlated to define trap generation conditions.

Accordingly, there is an effect that an object can be added or deleted as necessary, and especially, trap generation can be arbitrarily controlled.

In addition, the periodical management behavior does not performed for the object of which state is not changed, so that a traffic of the management behavior can be reduced.

Moreover, the agent is controlled and the state change is monitored by using the SNMP having the trap ON/OFF fields by objects, so that management efficiency in the management network can be increased.

As the present invention may be embodied in several forms without departing from the spirit or essential characteristics thereof, it should also be understood that the above-described embodiments are not limited by any of the details of the foregoing description, unless otherwise specified, but rather should be construed broadly within its spirit and scope as defined in the appended claims, and therefore all changes and modifications that fall within the meets and bounds of the claims, or equivalence of such meets and bounds are therefore intended to be embraced by the appended claims.